



## UWS Academic Portal

### Virtual observation of virtual reality simulations

Howie, Scott Ronald; Gilardi, Marco

*Published in:*

CHI EA 2019 - Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems

*DOI:*

[10.1145/3290607.3312836](https://doi.org/10.1145/3290607.3312836)

Published: 02/05/2019

*Document Version*

Peer reviewed version

[Link to publication on the UWS Academic Portal](#)

*Citation for published version (APA):*

Howie, S. R., & Gilardi, M. (2019). Virtual observation of virtual reality simulations. In *CHI EA 2019 - Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems: Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems* [LBW2710] Association for Computing Machinery. <https://doi.org/10.1145/3290607.3312836>

#### General rights

Copyright and moral rights for the publications made accessible in the UWS Academic Portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

#### Take down policy

If you believe that this document breaches copyright please contact [pure@uws.ac.uk](mailto:pure@uws.ac.uk) providing details, and we will remove access to the work immediately and investigate your claim.

---

# Virtual Observation of Virtual Reality Simulations

**Scott Ronald Howie**

University of the West of Scotland  
Paisley, United Kingdom, UK  
Scott.Howie@uws.ac.uk

**Marco Gilardi**

University of the West of Scotland  
Paisley, United Kingdom, UK  
Marco.Gilardi@uws.ac.uk

## ABSTRACT

Unlike conventional desktop simulations which have constrained interaction, immersive Virtual Reality (VR) allows users to freely move and interact with objects. In this paper we discuss a work-in-progress system that ‘virtually’ records participants movement and actions within a simulation. This system recovers and rebuilds recorded data on request, accurately replaying individual participants motions and actions in the simulation. Observers can review this reconstruction using an unrestricted virtual camera and if necessary, observe changes from recorded input devices. Reconstruction of each participants’ skeleton structure was created using tracked input devices. We conclude that our system offers detailed recreation of high-level knowledge and visual information of participant actions during simulations.

## KEYWORDS

Virtual Reality; VR; Observing; Replaying; Reviewing; Simulations; Virtual Observation

## INTRODUCTION

Reviewing VR simulations allows insight into participant performance [5] and determine the cause and effect relationships from participant actions [5]. Unlike desktop computers, VR equipment has

---

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

*CHI’19 Extended Abstracts, May 4–9, 2019, Glasgow, Scotland Uk*

© 2019 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-5971-9/19/05.

<https://doi.org/10.1145/3290607.3312836>



**Figure 1: Screenshot capture from video comparison showing real-time video capture of a simulation and rebuilt capture using Virtual Observation. The full video can be viewed here: [youtu.be/NLGKC60qCGY](https://youtu.be/NLGKC60qCGY). The Virtual Observation system can be included in any SteamVR compatible Unity project by adding the VO component to the default VR camera rig setup and linking the input states of the developers/researchers interaction system (so the actions can be recorded and later replicated). Recorded data can be stored in XML or Json file formats and supports all VR headsets and tracked controllers that are compatible with SteamVR and the Unity game engine.**

six-degrees of freedom [8]. Because of VR's unrestricted movement, usability issues are common for inexperienced users [8]. To observe participants using VR an observer would need to be present to take notes or record the participant. This type of observation is difficult, requires large data storage, can impact the simulation performance and offers limited insight of the participants experience. Experiencing the exact perspective of participants induces motion sickness on the observer [7]. Observations can be conducted from mirrored perspectives of VR participants on 2D-screens. But these recordings can not guarantee knowledge of the state and location of the input devices when they are not visible limiting studies outside laboratory conditions. Video recording participants offers a visual clarity of their movement in VR, but are fixed in position and require to be paired and synced with other data to be of any use.

This paper details an initial validation phase on our work-in-progress into virtually observing participants in VR simulations. The term 'Virtual Observation' (VO) is used as we are not observing the participant themselves, instead we observe the input movement and actions conducted by the user during the simulation. Collected input and movement data is used to re-create their exact experience in the simulation with synced actions and approximate movement. In the following sections we discuss our 'Virtual Observation' system and review the re-created simulations.

## RELATED WORK

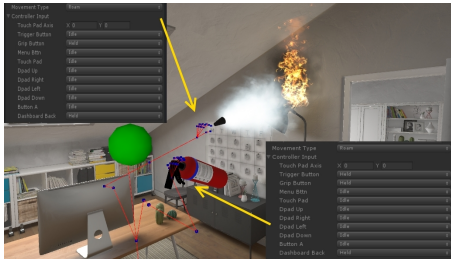
Research interest to review VR simulations is not new. Goldberg, Knerr and Grosse [4] used early implementations of VR technology to replay and review army training simulations in an effort to measure participants performance. More recently Lopez et al. [7] used a technique that creates and stores animation of movement and object manipulations conducted by the participant. Lopez et al.'s [7] system is restricted at re-creating the motions of the simulation, lacking high-level knowledge (user input or device information) of functionary events or actions. Without high-level knowledge it is difficult to detect usability issues in an interaction system.

Jung et al. [6] also developed a system that aims at creating re-usable animations for animating 3D character models. Jung et al.'s [6] recorded data posture combines the motion of the participant from tracking devices and pre-determined input states for grasping an identified object. Like Lopez et al. [7], Jung et al.'s [6] pre-defined input and interaction objects limit the high-level information that can be gathered from observing the animations.

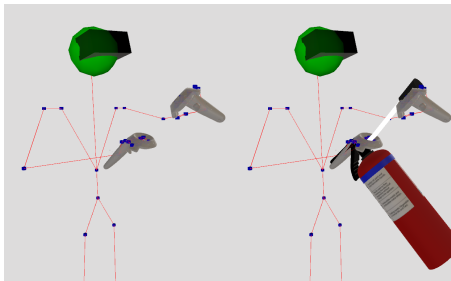
Both Lopez et al. [7] and Jung et al.'s [6] methods are limited and for in-depth analysis of participants actions in a simulation high-level data (input and states) is necessary.

## THE VIRTUAL OBSERVATION SYSTEM

'Virtual Observation' (VO) combines unrestricted observation of participants [3] with a form of action capture [6, 7] allowing participants to be observed from any perspective. Rather than storing



**Figure 2: Rebuilding the simulation from recorded data using the Unity Engine (only engine supported by the VO system at the moment). States are derived from this data and determine how the participant interacted with the virtual environment. Input states are overlaid from the Unity UI for visual clarity.**



**Figure 3: Screenshot of a participant in a default standing pose putting out a fire in the rebuilt simulation (see Figure 2 for alternative angle). (left) Participant skeleton pose. (right) Participant skeleton pose with held items.**

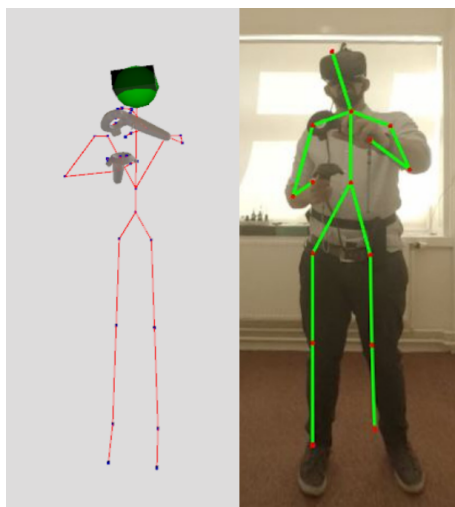
information of an entire environment, VO records participants movement and input changes. The major contribution of our VO system is that unlike [6] and [7], which record the motion of interacted virtual objects, only participants inputs and tracking information is recorded by the system. The motion of the virtual object participants interact with is reconstructed by our VO system from the participants input. VO allows individual simulations to be replayed, giving an insight on how exactly the participant acted during the simulation. However, any object that has unpredictable movement, such as a non-player controlled character or sensitive physics affected objects, may not be the same during the re-built simulation. To address this, any unpredictable object can be tracked separately. Positional and rotational data of the HMD and controllers is recorded at a fixed frame rate interval of 100 milliseconds (which can be modified by altering the refresh rate of the game engine). Virtual Observation must be integrated into the application during the development cycle prior to release. When a modified input state is detected, an input action key is recorded, which stores the motion of the controllers and the VR manager with the current input modifications. During the simulation VR configuration data is stored to allow for tracking play-space and headset information to be analysed. Rebuilt simulations use participant data-sets to replicate the movement and actions of the input devices, see Figure 2. Participant movement is smoothed between action and movement frames capture points to animate the movement of the controllers and HMD. During instances of input changes, the position and rotation of the controllers is forcibly set to ensure that interaction is correctly mapped at the exact position and orientation recorded and is not affected by any delays or gaps in the animation smoothing process.

Using Unity's Inverse Kinematic (IK) system, an estimated skeleton posture of the participant is created using the rebuilt tracking points from the VR equipment, see Figure 3, these tracking points map the head and hand locations of the participant to an avatar for real-time animation.

To observe participants, the observer controls a 3D virtual camera to explore the virtual environment. This allows to observe from any view the participants re-constructed actions and device motions during the replay. To reproduce a participant simulation the rendering and update of the Unity game engine had to be checked to prevent positional data from being set or timed incorrectly from the original data time-stamps. This required to modify the fixed update cycle of the Unity engine which we set to update every 0.01 seconds. The simulation should be assessed to ensure that frame intervals are acceptable for the needs of the simulation recording.

## METHODOLOGY

To evaluate our 'Virtual Observation' system we created a fire training simulation that aims to teach participants safety protocols. After obtaining ethical approval, we conducted a study with 12 participants in our lab controlled environment using an HTC Vive with a wireless TPCast attachment. Two GoPro Hero 4 cameras were setup at opposite areas of the controlled VR play-space area to



**Figure 4: Side by side comparison of an estimated skeleton posture obtained from our system (left) and a participants posture from GoPro video footage with OpenCV's pose estimation (right). The accuracy of the estimated skeleton posture will deteriorate as and when the participant moves untracked joints of their body. In our case, the lack of torso tracking caused the entire structure to rotate to match that of the HMD. Because feet and knee joints were untracked the IK system estimated the position of the knee joints relative to the ground and height of the HMD. Therefore, if the participant moved their feet from the default standing pose as seen above, the lower parts of the body would decrease in accuracy.**

video record a complete view of the tracked VR play-space and participant. Desktop screen-capture of participants perspective during the simulation was recorded at pre-determined interaction sections. In the training simulation participants were scored summatively on their performance from a grading system that assessed objectives completion.

Tracked motion and input data was instantly stored online making data available for analysis if a participant left or stopped the study early, offering a potential insight on what they did prior to exiting the application. A training stage familiarised participants with controls and the necessary safety protocols. After successfully completing all training tasks, participants performed a search task in a different virtual environment aiming at locating and extinguishing fires.

To rebuild the simulation, participants data were loaded using the unique ID code, allowing to replicate participants use of the input devices and to rebuild the simulation as experienced by the participant. To assess and validate our virtual observational software, five frames were identified for each participant (see Table 1). These frames were guaranteed points of interaction that were completed by all participants. Using Arbib's [1] action equation, we assessed simulation rebuilding accuracy by comparing similarities between the actual participant recordings and rebuilt recording.

Assessment of the skeleton posture was conducted by comparing screen-shots from when the actions were started as indicated by the input state. GoPro footage was synced to match simulation time with screen-shots taken and processed using OpenCV's PoseEstimation with default settings to estimate each participants skeleton posture (see Figure 4), which was used as the valid structural target. These OpenCV estimations were then compared to same frame screen-shots from our estimation IK software. Joints were classified valid (1) if their relative positioning from connected joints was structurally similar to the OpenCV joint for position and bend. Miss-aligned joints on the Unity IK structure were marked as invalid (0). Percentages were calculated by adding all valid joints for each frame and dividing them by the total number of joints, giving the overall percentage for each frame joint. Any OpenCV skeleton posture that was visibly miss-calculated was removed (12 images were removed) from the analysis. Table 1 gives a scoring for the evaluation of each frame assessed.

**Table 1: Averaged percentage values of skeleton estimation from comparing OpenCV's reconstructions from GoPro frames and estimated pose from VO. Frames were captured by the registered instance of an action being induced. The following frames were selected for comparison since these interactions were completed by all participants: 1. Opening garage door. 2. Opening house door. 3. Picking up fire extinguisher. 4. Extinguishing fire. 5. Pressing review simulation button. A high-accuracy value indicates that estimated joints were almost identical visually. Forty-eight images were compared as part of the analysis (twelve were removed as OpenCV failed to calculate a skeleton posture from the GoPro capture). Lack of torso tracking decreased the accuracy of the estimated IK skeleton structure, causing joint estimations to decrease in accuracy the further the joint was from a tracked point. However, because participants often assumed a default stance (Figure 4), the lack of feet or torso tracking was mostly negligible, but would deteriorated for participants whom varied from this stance.**

Frame	1	2	3	4	5
Torso	88%	96%	94%	100%	92%
L-Arm	94%	94%	90%	100%	94%
R-Arm	96%	96%	96%	100%	98%
L-Knee	94%	96%	94%	98%	96%
R-Knee	92%	96%	96%	98%	96%

## FINDINGS

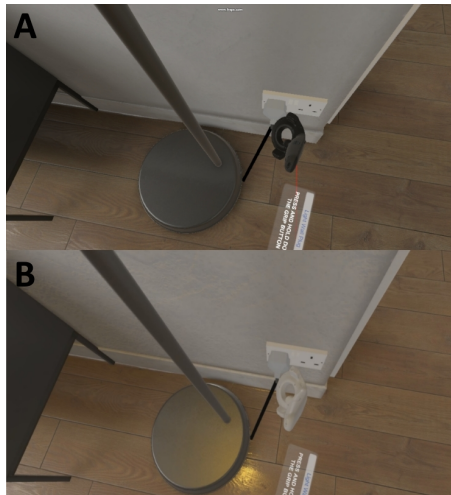
The ability to observe approximate poses and skeleton postures without restrictions offered insight into the motions of participants in the training simulation allowing for discovery of how and potentially why a user failed to complete an objective. We were able to identify numerous cases of participants failing to interact with objects and/or using the wrong input command/button. We were able to observe that fire extinguishers were frequently mishandled, likely produced by 'magical interaction' [2] which removes the senses of weight and cumbersomeness from real-life equipment. Participants also frequently attempted to interact with objects in incorrect positions.

From a developer standpoint, the ability to view and replay the exact input, actions and movement of participants offers the chance to directly observe the process that lead to bugs or errors. In our case, we noticed that if a participant activated the fire extinguisher but then dropped it, the fire extinguisher would continue to be active spraying foam. We also observed numerous cases of people failing to follow instructions and protocols for safety procedures. From our pre-defined scoring system alone, which looked at goal completion, we would have incorrectly assumed all participants followed safety protocols as instructed. Half of our participants would have failed the real-life counterpart training for improper handling of safety equipment or dangerous movement.

Accuracy of the skeleton posture, see Table 1, varied by stance. Head and hands were tracked accurately with data provided from the VR equipment. The posture of the estimated skeleton derived from the position and rotation of the headset which all other joint estimations relied upon. Due to the lack of torso tracking the IK system occasionally miss-calculated the locations of the arm and knee joints. Although the skeleton reconstruction is not perfect yet, the strength of this work-in-progress is its ability to reconstruct the actions of the participant during the simulation. A demonstration video (Figure 1) shows a comparison between the rebuilt simulation camera and the screen-capture recording. Developers and researchers can use VO to analyse feasibility and usability of systems. Examiners of training simulations can use the data to check and assess how participants have completed training procedures and verify they are correctly handling equipment/protocols. VO allows future comparison between participant performance or to assess if a bug present in the recording has been fixed. On average, each participant submitted a data-set of around 80MB for the simulation which took ten to fifteen minutes to complete. Data-set vary depending on participants actions and length in the simulation. In all cases, these data-sets were by large smaller than GoPro video recordings which were around 5.8GB for each participant and screen-recording of around 500MB for 30 seconds.

## CONCLUSION

This paper details our work-in-progress VO software. VO records data from VR simulations and reconstructs the experience using users actions and motions. Our findings indicate that rebuild



**Figure 5: (top) Screenshot capture from screen-recording of participant. (bottom) Screenshot of the HMD FoV of the rebuilt simulation from the same participant.**

simulations were almost identical to the original simulation (see Figure 5). We conclude that these recordings are on par with [7], but offer greater insight into high-level technical details of the simulation (how the actions of the user affect the simulation procedure), offering context from cause and effect relationships of participants actions [5].

Using only two controllers and a VR headset as tracking devices, the reconstructed skeleton structure of participants varied in quality depending on the posture of the participant. Lack of torso tracking often limited accurate estimation of the body orientation and stance.

Lack of finger tracking on the controller limits the observation of controllers usability as non-registered input on the controllers can not be tracked.

We conclude that virtual observation of reconstructed simulations is a highly effective and versatile tool for researchers and developers, with reconstructed participant actions almost identical to real performance. In the future an ‘in the wild’ study will be conducted to assess the viability of virtually observing participants outside of laboratory conditions. Further on this research will be expanded to determine if and how such a system could be incorporated directly into a evaluation system that provides instant feedback [5]. Additional tracking devices and features (eye-tracking, torso, audio) will be included to improve data acquisition.

## REFERENCES

- [1] Michael A Arbib. 2002. The Mirror System, Imitation, and the Evolution of Language. *Imitation in animals and artifacts* 229 (2002).
- [2] Doug A Bowman, Ryan P McMahan, and Eric D Ragan. 2012. Questioning naturalism in 3D user interfaces. *Commun. ACM* 55, 9 (2012), 78–88.
- [3] Joel Carranza, Christian Theobalt, Marcus A Magnor, and Hans-Peter Seidel. 2003. Free-viewpoint video of human actors. In *ACM transactions on graphics (TOG)*, Vol. 22. ACM, 569–577.
- [4] Stephen L Goldberg, Bruce W Knerr, and James Grosse. 2003. *Training dismounted combatants in virtual environments*. Technical Report. Army Research Development and Engineering Command Orlando FL Simulation.
- [5] Samer Hanoun and Saeid Nahavandi. 2018. Current and future methodologies of after action review in simulation-based training. In *Proceeding of the 2018 Annual IEEE International Systems Conference (SysCon)*. IEEE, 1–6.
- [6] Bernhard Jung, Heni Ben Amor, Guido Heumer, and Matthias Weber. 2006. From motion capture to action capture: A review of imitation learning techniques and their application to VR-based character animation. In *Proceedings of the ACM symposium on Virtual reality software and technology*. ACM, 145–154.
- [7] Thomas Lopez, Olivier Dumas, Fabien Danieau, Bertrand Leroy, Nicolas Mollet, and Jean-François Vial. 2017. A playback tool for reviewing VR experiences. In *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology*. ACM, 83.
- [8] Jolanda G Tromp, Anthony Steed, and John R Wilson. 2003. Systematic usability evaluation and design issues for collaborative virtual environments. *Presence: Teleoperators & Virtual Environments* 12, 3 (2003), 241–267.